UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/672,248	09/29/2003	Monika Henzinger	0026-0043	8702

44989 7590 11/29/2005

HARRITY & SNYDER, LLP
11240 WAPLES MILL ROAD
SUITE 300
FAIRFAX, VA 22030

EXAMINER	
RUTLEDGE, AMELIA L	
ART UNIT	**PAPER NUMBER**
2176	

DATE MAILED: 11/29/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/672,248	HENZINGER ET AL.	
	Examiner	Art Unit	
	Amelia Rutledge	2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 September 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 September 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *1/29/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to communications: original application, filed 09/29/2003.

2. Claims 1-31 are pending in the case. Claims 1, 10, 15, 21, 26, and 31 are independent claims.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1-8 and 31 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5. **Regarding independent claims 1 and 31**, the term " *based, at least in part,* " in claims 1 and 31 is a relative term which renders the claim indefinite. The term " *based, at least in part,* " is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

6. **Regarding dependent claims 2-8,** claims 2-8 are rejected for fully incorporating the deficiencies of their respective base claim.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. **Claims 1, 2, and 9-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over DaCosta et al. (hereinafter "DaCosta"), U.S. Patent No. 6,665,658, issued December 2003, in view of Loke et al. (hereinafter "Loke"), "Logic Programming with the World-Wide Web", p. 235-245, *Hypertext 1996*, ACM.**

Independent claim 1 cites: *A method for crawling documents comprising: receiving a uniform resource locator (URL), and determining whether the URL is associated with a web site that uses session identifiers based, at least in part, on a comparison of a portion of URLS that change between different copies of a document downloaded from the web site.*

DaCosta teaches a method for crawling documents in a dynamic website, with a database for storing and identifying session identifiers URLs, and an application program for controlling a software agent (Col. 4, l. 41-Col. 5, l. 23). DaCosta teaches the analysis of URLs and headers to determine if a web site uses session IDs (Col. 6, l. 21-40). Loke teaches the use of URLs with session identifiers which can be extracted when required (p. 4, par. 47-57). While DaCosta teaches the use of a software agent which determines session data in the process of crawling the documents on the web site, Loke teaches the use of structured logic programming for various objectives in crawling a web site (p. 238-239, "A Page Searcher Application). Loke also teaches the

use of logic to compare URLs and attach state to URLs, with clauses that handle clause insertions and deletions into the state and clauses that attempt to prove goals using the state and the module (p. 239, "Using the Notion of State"). The logic programming taught by Loke would allow the comparison of URLs of downloaded document copies for determining whether the URL is associated with session identifiers.

Both DaCosta and Loke are directed toward controlling software search agents and tracking state and sessions. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Loke to DaCosta, so that the user would have the benefit of mapping a web page to a logic program so that the page would be enabled to reason about itself and other pages and define arbitrary relationships between pages (Loke, p. 235, par. 6).

Regarding dependent claim 2, DaCosta teaches that the method of analyzing session ids can be applied to any web page in a site (Col. 6, l. 20-40), compare to *wherein the document is a home page of the web site.*

Regarding dependent claim 9, DaCosta does not explicitly teach analyzing URLs to determine whether the portion of URLs that change are greater than a predetermined value, but DaCosta does teach analyzing URLs to determine whether a page or site uses session identifiers. Loke teaches a comparison of URLs using logic rules (p. 238-239, "A Page Searcher Application) compare to *wherein the comparison determines that the web site uses session identifiers when the portion of the URLS that change is greater than a predetermined value.* It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the method of scanning

and analyzing session IDs taught by DaCosta with the URL string comparisons taught by Loke to determine whether a site uses session identifiers by applying logic rules to the URL string. Both DaCosta and Loke are directed toward controlling software search agents and tracking state and sessions. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Loke to DaCosta, so that the user would have the benefit of mapping a web page to a logic program so that the page would be enabled to reason about itself and other pages and define arbitrary relationships between pages (Loke, p. 235, par. 6).

Independent claim 10 cites: *A method for identifying web sites that use session identifiers comprising: downloading at least two different copies of at least one document from a web site; extracting uniform resource locators (URLs) from the two different copies of the web document; comparing the extracted URLS of the two different copies of the document; and determining whether the web site uses session identifiers based on the comparison.*

DaCosta teaches a method for crawling documents in a dynamic website, with a database for storing and identifying session identifiers URLs, and an application program for controlling a software agent (Col. 4, l. 41-Col. 5, l. 23). DaCosta teaches the analysis of URLs and headers to determine if a web site uses session IDs (Col. 6, l. 21-40). The analyzed URLs are stored in the site database. DaCosta teaches the identification of duplicated content (Col. 8, l. 12-15). While DaCosta does not explicitly teach comparing the extracted URLs of two different copies of the document and determining whether the site uses session identifiers based on the comparison, Loke

teaches the use of structured logic programming for various objectives in crawling a web site (p. 238-239, "A Page Searcher Application). Loke also teaches the use of logic to compare URLs and attach state to URLs, with clauses that handle clause insertions and deletions into the state and clauses that attempt to prove goals using the state and the module (p. 239, "Using the Notion of State"). The logic programming taught by Loke would allow the comparison of URLs of downloaded document copies for determining whether the URL is associated with session identifiers. Both DaCosta and Loke are directed toward controlling software search agents and tracking state and sessions. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Loke to DaCosta, so that the user would have the benefit of mapping a web page to a logic program so that the page would be enabled to reason about itself and other pages and define arbitrary relationships between pages (Loke, p. 235, par. 6).

Regarding dependent claim 11, DaCosta does not explicitly teach determining that the web site uses session identifiers when the comparison indicates that at least a predetermined portion of the URLS change between the two different copies, but DaCosta does teach analyzing URLs to determine whether a page or site uses session identifiers. Loke teaches a comparison of URLs using logic rules (p. 238-239, "A Page Searcher Application). Loke also teaches proofs using logic, i.e., determining goals by comparisons with a predetermined value (p. 240, "Structured LP"). Further, this type of comparison was notoriously well known in the art at the time of the invention. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the method of scanning and analyzing session IDs taught by DaCosta with the

URL string comparisons taught by Loke to determine whether a site uses session identifiers by applying logic rules to the URL string. Both DaCosta and Loke are directed toward controlling software search agents and tracking state and sessions. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Loke to DaCosta, so that the user would have the benefit of mapping a web page to a logic program so that the page would be enabled to reason about itself and other pages and define arbitrary relationships between pages (Loke, p. 235, par. 6).

Regarding dependent claim 12, DaCosta teaches extracting URLs local to a web site (Claim 1).

Regarding dependent claim 13, claim 13 reflects substantially similar subject matter as claimed in dependent claim 2, and is rejected along the same rationale.

Regarding dependent claim 14, while DaCosta teaches the analysis of session IDs, DaCosta does not explicitly teach using rules. Loke teaches the application of heuristics, i.e., the automatic application of rules. Both DaCosta and Loke are directed toward controlling software search agents and tracking state and sessions. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Loke to DaCosta, so that the user would have the benefit of mapping a web page to a logic program so that the page would be enabled to reason about itself and other pages and define arbitrary relationships between pages (Loke, p. 235, par. 6).

Independent claim 15 cites: *A device comprising: a spider component configured to crawl web documents associated with at least one web site; and a session identifier component configured to determine whether the web site uses session*

identifiers based on a comparison of a portion of uniform resource locators (URLS) that change between different copies of at least one web document downloaded from the web site.

DaCosta teaches a method for crawling documents in a dynamic website, with a database for storing and identifying session identifiers URLs, and an application program for controlling a software agent (Col. 4, l. 41-Col. 5, l. 23). DaCosta teaches the analysis of URLs and headers to determine if a web site uses session IDs (Col. 6, l. 21-40). DaCosta teaches a spider component (Col. 5, l. 49-65) and a session identifier component (Col. 6, l. 21-40). While DaCosta does note explicitly teach a comparison of a portion of uniform resource locators (URLS) that change between different copies of at least one web document, Loke teaches the use of structured logic programming for various objectives in crawling a web site (p. 238-239, "A Page Searcher Application). Loke also teaches the use of logic to compare URLs and attach state to URLs, with clauses that handle clause insertions and deletions into the state and clauses that attempt to prove goals using the state and the module (p. 239, "Using the Notion of State"). The logic programming taught by Loke would allow the comparison of URLs of downloaded document copies for determining whether the URL is associated with session identifiers. Both DaCosta and Loke are directed toward controlling software search agents and tracking state and sessions. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Loke to DaCosta, so that the user would have the benefit of mapping a web page to a logic program so that the page

would be enabled to reason about itself and other pages and define arbitrary relationships between pages (Loke, p. 235, par. 6).

Regarding dependent claim 16, DaCosta teaches that the spider component has a fetch component for downloading content, i.e., a requester (Col. 7, l. 25-25) and a content manager to extract URLs (Col. 7, l. 35-49), compare to *fetch component configured to download content from a network; and a content manager configured to extract URLS from the downloaded content.*

Regarding dependent claim 17, DaCosta teaches a site information database to store extracted URLs (Claim 5).

Regarding dependent claim 18, claim 18 reflects substantially similar subject matter as claimed in dependent claim 2, and is rejected along the same rationale.

Regarding dependent claim 19, DaCosta teaches tracking session IDs in URLs local to a web site (Col. 6, l. 20-40).

Regarding dependent claim 20, while DaCosta does teach the analysis of URLs for session IDs, DaCosta does not explicitly teach a rule generator. However, Loke teaches a comparison of URLs using logic rules (p. 238-239, "A Page Searcher Application). Both DaCosta and Loke are directed toward controlling software search agents and tracking state and sessions. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Loke to DaCosta, so that the user would have the benefit of mapping a web page to a logic program so that the page would be enabled to reason about itself and other pages and define arbitrary relationships between pages (Loke, p. 235, par. 6).

Regarding independent claim 21, claim 21 is directed toward the device used for implementing the method as claimed in claim 10, and is rejected along the same rationale.

Regarding dependent claims 22 and 23, claims 22 and 23 reflect substantially similar subject matter as claimed in dependent claims 11 and 12, and are rejected along the same rationale.

Regarding dependent claim 24, claim 24 reflects substantially similar subject matter as claimed in dependent claim 2, and is rejected along the same rationale.

Regarding dependent claim 25, claim 25 reflects substantially similar subject matter as claimed in dependent claim 14, and is rejected along the same rationale.

Regarding independent claim 26, claim 26 is directed toward the computer-readable medium containing programming instruction for executing the method as claimed in claim 10, and is rejected along the same rationale.

Regarding dependent claims 27 and 28, claims 27 and 28 reflect substantially similar subject matter as claimed in dependent claims 11 and 12, and are rejected along the same rationale.

Regarding dependent claim 29, claim 29 reflects substantially similar subject matter as claimed in dependent claim 2, and is rejected along the same rationale.

Regarding dependent claim 30, claim 30 reflects substantially similar subject matter as claimed in dependent claim 14, and is rejected along the same rationale.

Independent claim 31 cites: *A method for crawling documents comprising: receiving a uniform resource locator (URL), and determining whether the URL is*

associated with a web site that uses session identifiers based, at least in part, on a comparison of content between different duplicate or near-duplicate copies of a document downloaded from the web site for two different URLS.

DaCosta teaches a method for crawling documents in a dynamic website, with a database for storing and identifying session identifiers URLs, and an application program for controlling a software agent (Col. 4, l. 41-Col. 5, l. 23). DaCosta teaches the analysis of URLs and headers to determine if a web site uses session IDs (Col. 6, l. 21-40). The analyzed URLs are stored in the site database. DaCosta teaches the identification of duplicated content (Col. 8, l. 12-15). While DaCosta does not explicitly teach a comparison of content between different duplicate or near-duplicate copies of a document downloaded from the web site for two different URLS, Loke teaches the use of structured logic programming for various objectives in crawling a web site (p. 238-239, "A Page Searcher Application). Loke also teaches the use of logic to compare URLs and attach state to URLs, with clauses that handle clause insertions and deletions into the state and clauses that attempt to prove goals using the state and the module (p. 239, "Using the Notion of State"). The logic programming taught by Loke would allow the comparison of URLs of downloaded document copies for determining whether the URL is associated with session identifiers. Both DaCosta and Loke are directed toward controlling software search agents and tracking state and sessions. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Loke to DaCosta, so that the user would have the benefit of mapping a web page to a logic

program so that the page would be enabled to reason about itself and other pages and define arbitrary relationships between pages (Loke, p. 235, par. 6).

9. **Claims 3-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over DaCosta in view of Loke as applied to claims above, and further in view of Verity Ultraseek, Support FAQ #1037, created January 2002, p. 1-2 (hereinafter "Ultraseek").**

Regarding dependent claim 3, while DaCosta in view of Loke does teach extracting session IDs from cookies, DaCosta in view of Loke does not explicitly teach extracting a session identifier from the URL to obtain a clean URL. However, Ultraseek teaches extracting a session ID from the URL (p. 1-2). Loke teaches determining logic information based on a comparison of URLs by state (p. 239, "Using the Notion of State") and other logic operations used to compare URLs, compare to *determining whether the URL has already been crawled based, at least in part, on a comparison of the clean URL to a set of clean URLS that represent previously crawled URLS.* DaCosta, Loke, and Ultraseek are directed toward software search programs. It would have been obvious to one of ordinary skill in the art to combine Ultraseek, DaCosta, and Loke, because URL rewriting was a very common programming practice in the art at the time of the invention.

Regarding dependent claim 4, DaCosta teaches tracking session IDs in URLs local to a web site (Col. 6, l. 20-40), compare to *the portion of the URLS that change are identified using URLS that are local to the web site.*

Regarding dependent claims 5 and 6, while DaCosta teaches the analysis of session IDs, DaCosta does not explicitly teach using rules. Loke teaches the application of heuristics, i.e., the automatic application of rules, compare to *wherein the session identifiers from the URLS are extracted using rules for the web site.* DaCosta, Loke, and Ultraseek are directed toward software search programs. It would have been obvious to one of ordinary skill in the art to combine Ultraseek, DaCosta, and Loke, because URL rewriting was a very common programming practice in the art at the time of the invention.

Regarding dependent claim 7, DaCosta teaches that the URL is received as a URL from a previously crawled web document, because DaCosta teaches a method using URL marking of whether the site is interactive or not, for example (Col. 6, l. 20-40).

Regarding dependent claim 8, while DaCosta does not explicitly teach crawling the URL when the URL is determined to not already have been crawled, Loke teaches the use of logic to compare URLs and attach state to URLs, with clauses that handle clause insertions and deletions into the state and clauses that attempt to prove goals using the state and the module (p. 239, "Using the Notion of State"). The attachment of state to URLs would allow the robot to crawl the URL by determining the state of the URL, i.e., whether it had already been crawled. DaCosta, Loke, and Ultraseek are directed toward software search programs. It would have been obvious to one of ordinary skill in the art to combine Ultraseek, DaCosta, and Loke, because URL

rewriting was a very common programming practice in the art at the time of the invention.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

DeRoure, et al., "Investigating Link Service Infrastructures", copyright 2000, ACM, p. 67-76.

Hughes et al. U.S. Pub. No. 2003/0018779 published January 2003

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Amelia Rutledge whose telephone number is 571-272-7508. The examiner can normally be reached on Monday - Friday 9:30 - 6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AR

William L Bashore
WILLIAM BASHORE
PRIMARY EXAMINER
11/23/2005